



From FinisherSC to Meta-FinisherSC : Tools to Upgrade *de novo* Assemblies Using Long Reads

Ka-Kit Lam¹, Richard Hall³, Asif Khalak³, Kurt LaButti², Lior Pachter¹, David Tse^{1,4}

¹Department of EECS, UC Berkeley; ²Joint Genome Institute; ³Pacific Biosciences; ⁴Department of Electrical Engineering, Stanford University

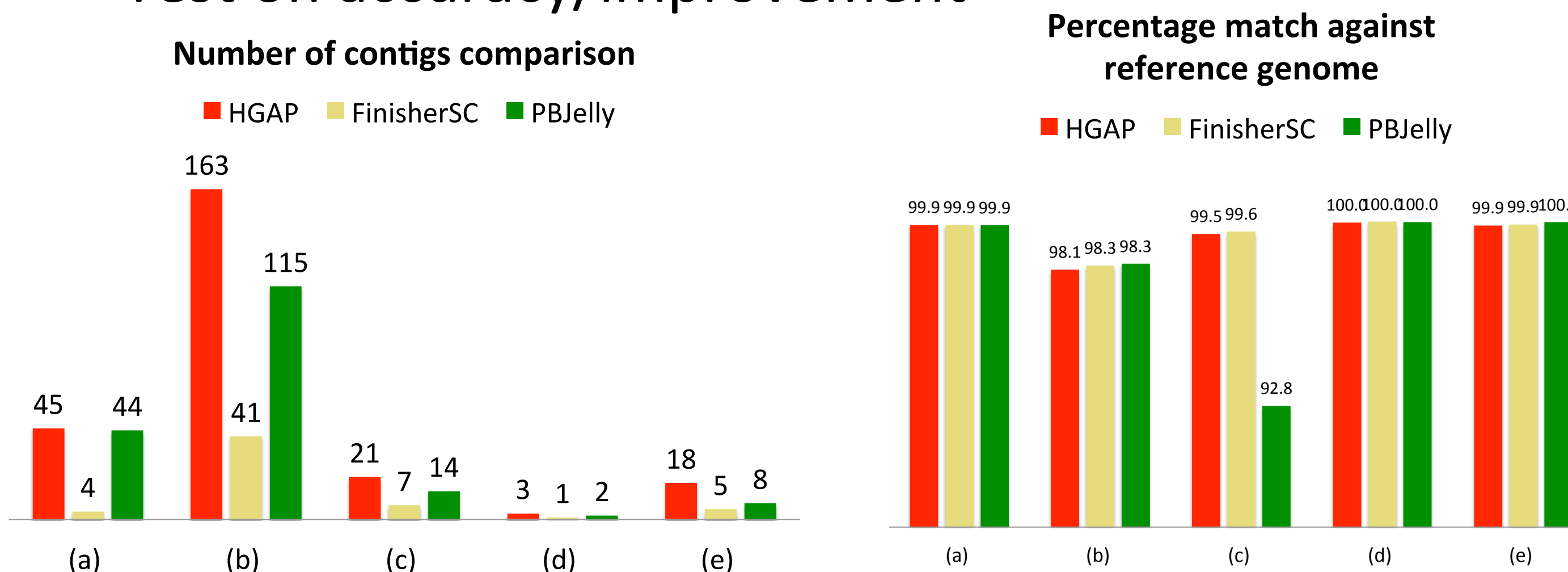
FinisherSC

Problem

- Long read data are trimmed and/or thrown away frequently.
- Can we improve the data efficiency of state-of-the-art long read *de novo* assembly pipelines for single genome shot-gun sequencing ?

Results [Lam, LaButti, Khalak, Tse, Bioinformatics 2015]

- Test on accuracy/improvement



Experimental evaluation results. (a,b) : *Pedobacter heparinus* DSM 2366 (long reads from JGI) (c, d, e) : *Escherichia coli* MG 1655, *Meiothermus ruber* DSM 1279, *Pedobacter heparinus* DSM 2366 (real long reads supporting the HGAP publication).

- Test on scalability

Genome name	Genome size (Mbp)	Size of reads (Gbp)	Running time (hours)
Caenorhabditis elegans	104	7.65	23
Drosophila	138	2.27	9.4
Saccharomyces cerevisiae	12.4	1.40	0.66

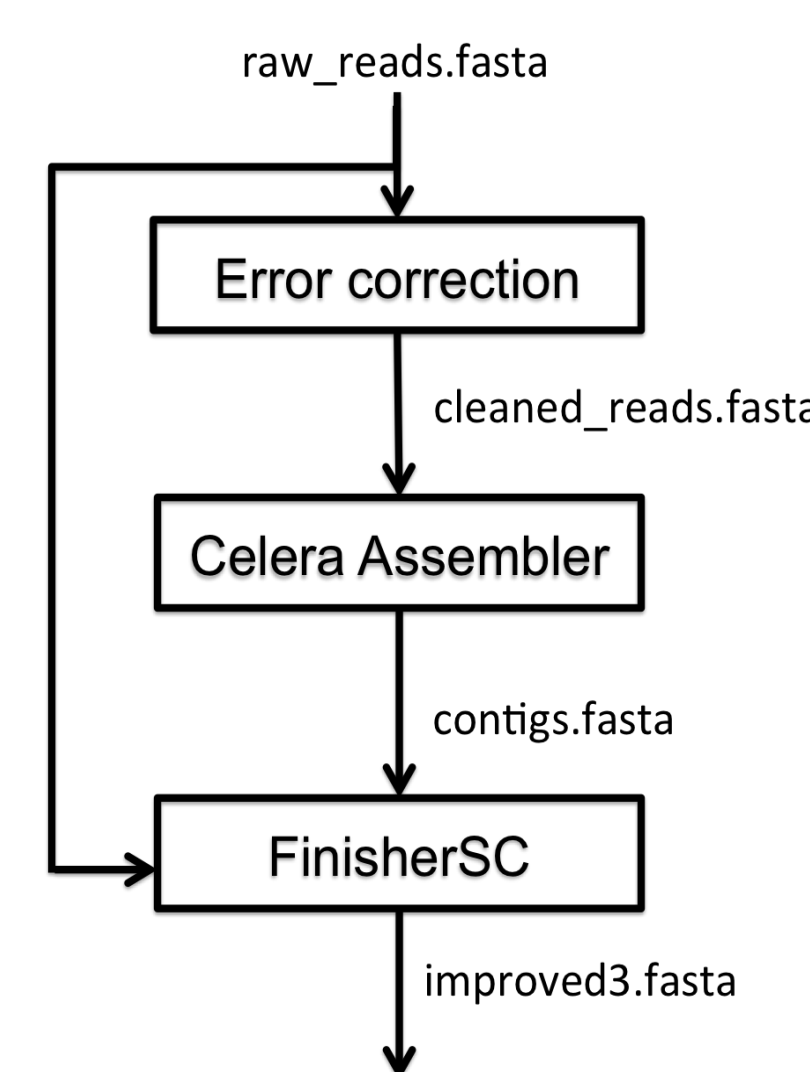
We run FinisherSC with the option of using 20 threads (-par 20) on a server computer. The server computer is equipped with 64 cores of CPU at clock rate of 2.4-3.3GHz and 512GB of RAM.

Methods

- Data-efficient :**
FinisherSC utilizes all the raw reads to perform re-layout.

- Repeat-aware :**
FinisherSC resolves repeats through operations on string graphs. Extensions (X-phaser and T-solver) resolve more complex repeats through multiple sequence alignment and copy count estimation.

- Scalable :**
FinisherSC streams raw reads and uses MUMmer for alignment.



Meta-FinisherSC-LongRead

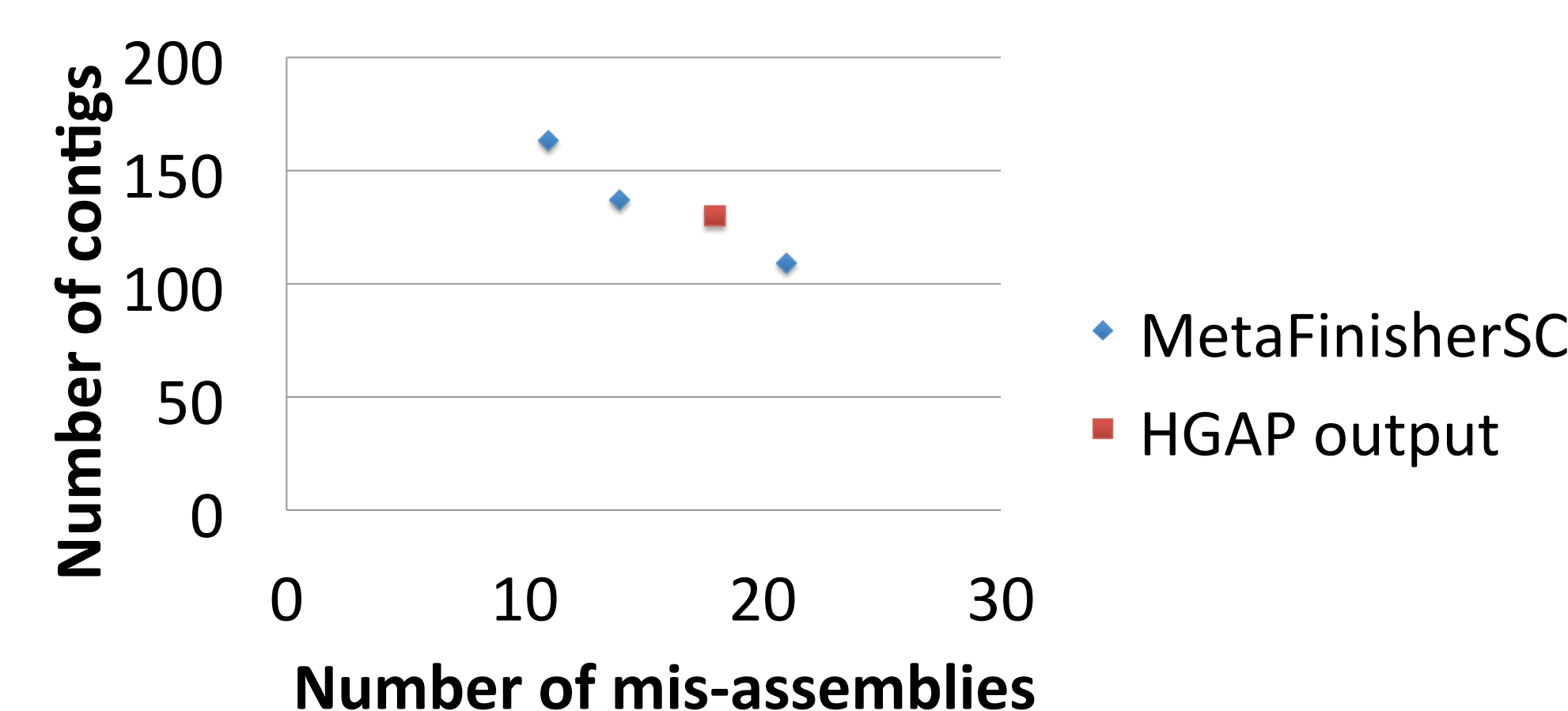
Problem

- In metagenomics shot-gun sequencing, genomes exist with diverse abundances and some genomes are very similar.
- Mis-assemblies are common and assemblies tend to be fragmented.
- Can we improve state-of-the-art long read *de novo* assembly pipelines for metagenomics shot-gun sequencing by reducing mis-assemblies and/or reducing fragmentation?

Results [manuscript in preparation]

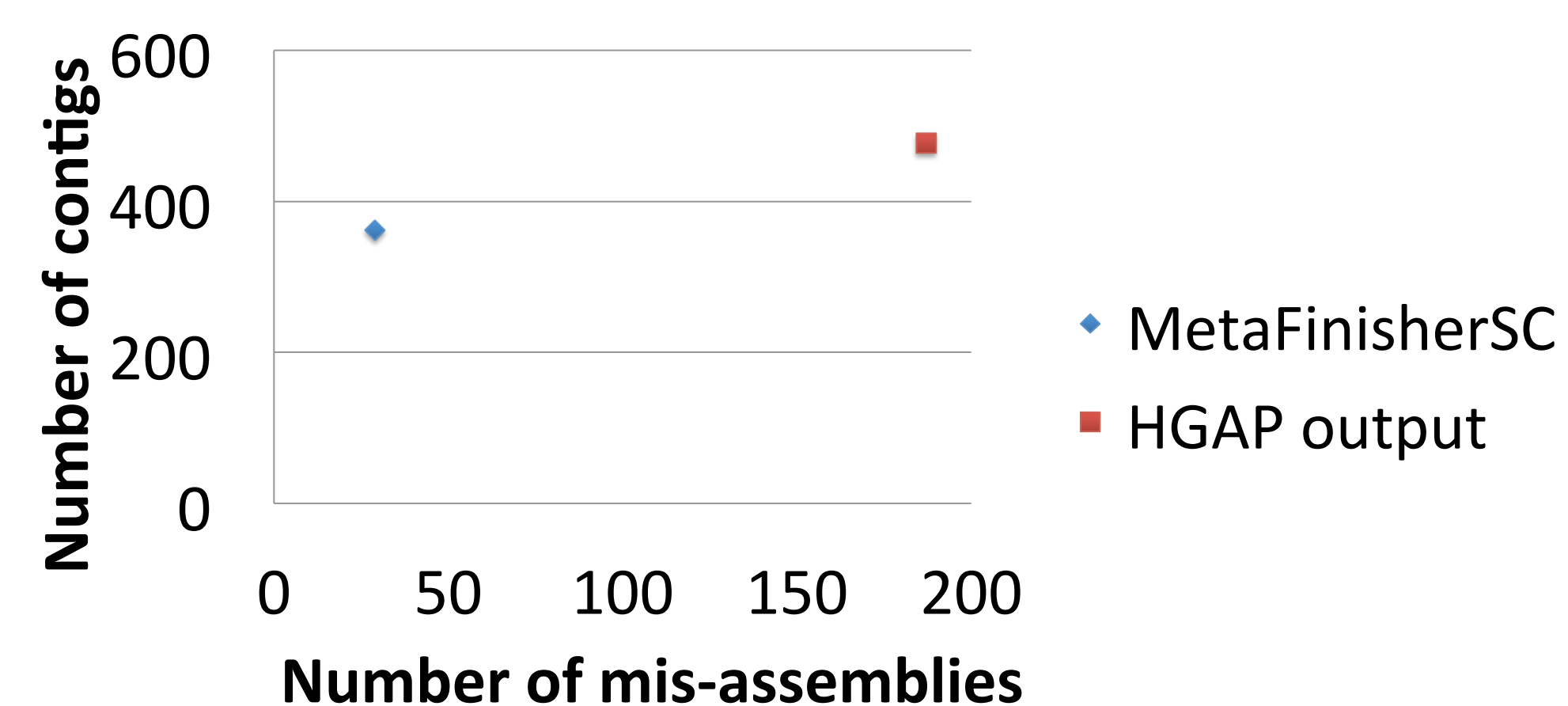
Example use case (fix mis-assemblies in the presence of very similar genomes):

- 5 species (*E. coli*, *Streptomyces A*, *Streptomyces B*, *C. difficile A*, *C. difficile B*) with two *Streptomyces* at ~80% sequence identity in similar regions, and two *C. difficile* at ~97% identity across the genome

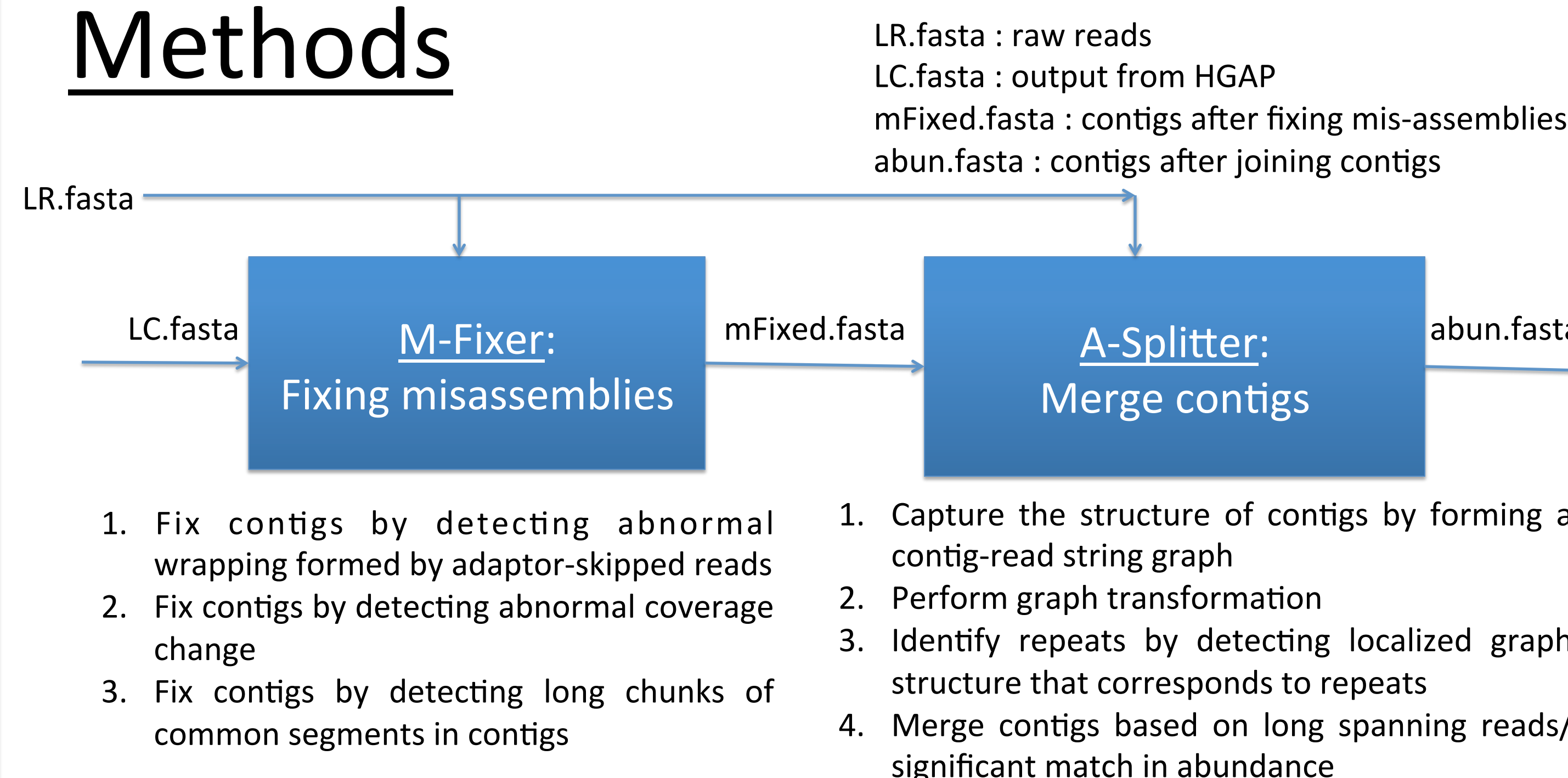


Example use case (improve assembly quality for low coverage data):

- 23 species with diverse abundances. Some are as low as 1X (BEI mock community)



Methods



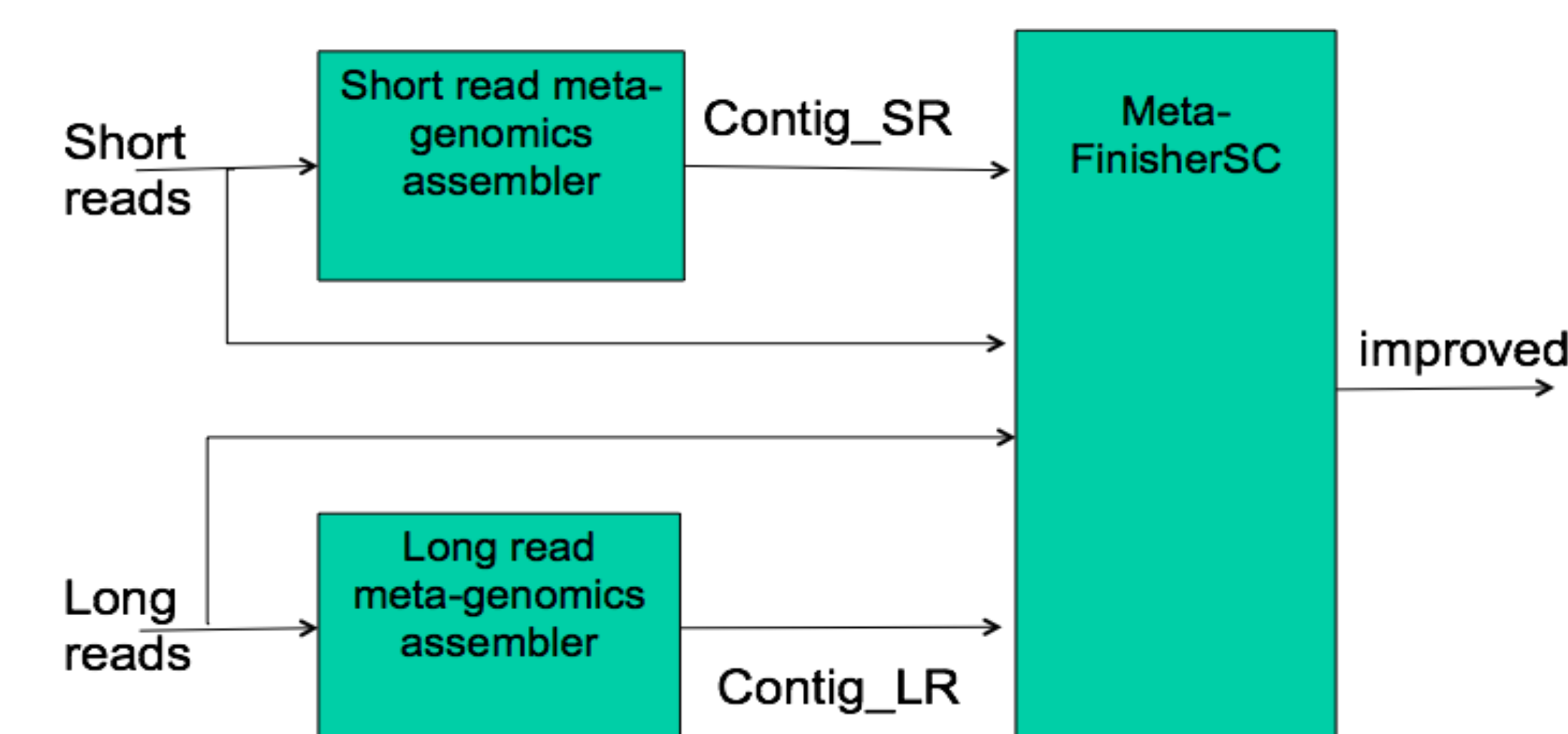
Meta-FinisherSC-MixedRead

Problem

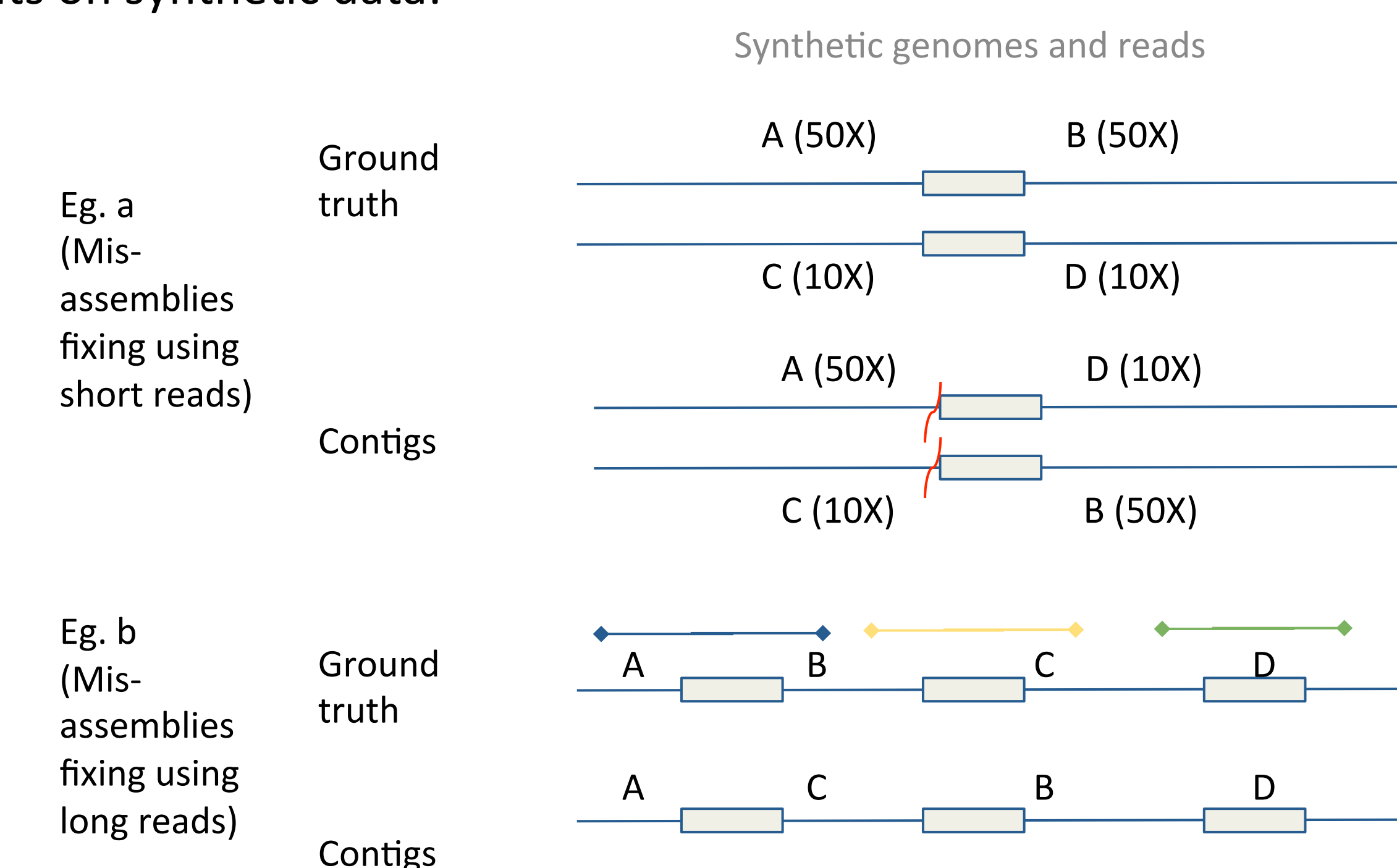
- It is common to have multiple data sources (short reads and long reads) when performing metagenomics assembly.
- Can we improve state-of-the-art *de novo* assembly pipelines for metagenomics shot-gun sequencing to seamlessly utilize all these data in a data-efficient manner?

Current prototype and future work

Pipeline:



Results on synthetic data:



Future work:

- Incorporate mate-pairs in the pipeline
- Scale up the system by MapReduce
- Design a unifying graph structure that can neatly combine all data sources
- Mathematically prove the performance guarantee of various components in the system

Acknowledgement

- We would like to thank Jason Chin(PacBio), Alicia Clum(JGI), James Drake (PacBio), Lorian Schaeffer(UC Berkeley), Lizzy Wilbanks(Caltech), S.M. Yiu (University of Hong Kong) for helpful discussion.

- This work is supported by the Center for Science of Information (CSol), an NSF Science and Technology Center, under grant agreement CCF-0939370.

** For more information/download on FinisherSC, please go to <http://kakitone.github.io/finishingTool/>

For more information/download on MetaFinisherSC, please go to <https://github.com/kakitone/MetaFinisherSC>